Ty Limpasuvan

ACM 104 Homework 4

① $p(x) = \|Ax-b\|^2 = x^T(A^TA)x - 2x^T(A^Tb) + \|b\|^2$

$$= x^TKx - 2xf + c$$

$$K = A^T \cdot A = \begin{bmatrix} 1 & 0 & 1 & -3 \\ 2 & -2 & 5 & 1 \\ -1 & 3 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & -1 \\ 0 & -2 & 3 \\ 1 & 5 & -1 \\ -3 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 11 & 4 & -5 \\ 4 & 34 & -12 \\ -5 & -12 & 12 \end{bmatrix}$$

$$f = A^Tb = \begin{bmatrix} 1 & 2 & -1 \\ 0 & -2 & 3 \\ 1 & 5 & -1 \\ -3 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 5 \\ 6 \\ 8 \end{bmatrix} = \begin{bmatrix} -18 \\ 28 \\ 17 \end{bmatrix}$$

Need to find if K is positive definte (if all of its principal minors are positive). $\det A_1 = 11$, $\det A_2 = 18$, $\det A_3 = 2342$, all of which are positive. Thus K is positive definite and there exists global minimizer

$$x^* = K^{-1}f$$

$$= Kf = \begin{bmatrix} 11 & 4 & -5 \\ 4 & 34 & -12 \\ -5 & -12 & 12 \end{bmatrix} \begin{bmatrix} -18 \\ 28 \\ 17 \end{bmatrix} = \frac{1}{2342} \begin{bmatrix} 264 & 12 & 122 \\ 12 & 107 & 112 \\ 122 & 112 & 358 \end{bmatrix} \begin{bmatrix} -18 \\ 28 \\ 17 \end{bmatrix}$$

$$= \begin{bmatrix} -1 \\ 2 \\ 3 \end{bmatrix} \leftarrow \text{Solution}$$

with LSE of

$$\sqrt{\|b\|^2 - b^TAx^*} = \sqrt{\sqrt{125}^2 - 125} = 0$$

which means x^* is an exact solution

② In parts Ⓐ and Ⓑ, $p_n(x)$ has degree 1 We can thus derive $p_1(x)$:

$$p_1(x) = f(x_0) \cdot L_1(x) + f(x_1) \cdot L_2(x) = f(x_0) \cdot \frac{x - x_1}{x_0 - x_1} + f(x_1) \cdot \frac{x - x_0}{x_1 - x_0}$$

$$= \frac{f(x_0)}{x_0 - x_1} x - \frac{x_1 f(x_0)}{x_0 - x_1} + \frac{f(x_1)}{x_1 - x_0} x - \frac{x_0 f(x_1)}{x_1 - x_0}$$

$$= \frac{f(x_0) - f(x_1)}{x_0 - x_1} x + \frac{x_0 f(x_1) - x_1 f(x_0)}{x_0 - x_1}$$

Ⓐ $x_0 = a$, $x_1 = b \rightsquigarrow p_1(x)$

$$p_1(x) = \frac{f(a) - f(b)}{a - b} + \frac{af(b) - b f(a)}{a - b}$$

$$\int_a^b f(x)dx \approx \int_a^b p_1(x)\, dx = \int_a^b \left(\frac{f(a) - f(b)}{a - b} x + \frac{af(b) - bf(a)}{a - b} \right) dx$$

$$= \left[\frac{f(a) - f(b)}{a - b} \cdot \frac{x^2}{2} + \frac{af(b) - bf(a)}{a - b} \cdot x \right]_a^b$$

$$= \frac{f(a) - f(b)}{a - b} \cdot \frac{b^2}{2} + \frac{af(b) - bf(a)}{a - b} b - \frac{f(a) - f(b)}{a - b} \cdot \frac{a^2}{2} - \frac{af(b) - bf(a)}{a - b} a$$

$$= \frac{f(a) - f(b)}{a - b} \cdot \frac{b^2 - a^2}{2} - af(b) + bf(a) = \frac{1}{2} (f(b) - f(a))(b + a) - af(b) + bf(a)$$

$$= \frac{1}{2} [bf(b) + af(b) - bf(a) - af(a) - 2af(b) + 2bf(a)]$$

$$= \frac{1}{2} [f(b) + f(a)] (b - a)$$

Ty Limpasuvan

(B) $x_0 = \frac{1}{3}(a+b)$, $x_1 = \frac{2}{3}(a+b) \rightsquigarrow p_1(x)$

$$p_1(x) = \frac{f(x_0) - f(x_1)}{\frac{1}{3}(a+b) - \frac{2}{3}(a+b)} x + \frac{\frac{1}{3}(a+b)f(x_1) - \frac{2}{3}(a+b)f(x_0)}{\frac{1}{3}(a+b) - \frac{2}{3}(a+b)}$$

$$= [f(x_1) - f(x_0)] \cdot \frac{3x}{a+b} + 2f(x_0) - f(x_1)$$

$$\int_a^b f(x)\,dx \approx \int_a^b p_1(x)\,dx = \int_a^b \left([f(x_1) - f(x_0)] \cdot \frac{3x}{a+b} + 2f(x_0) - f(x_1)\right) dx$$

$$= \left[\frac{3}{2}[f(x_1) - f(x_0)] \cdot \frac{x^2}{a+b} + 2xf(x_0) - xf(x_1)\right]_a^b$$

$$= \frac{3}{2}[f(x_1) - f(x_0)] \cdot \frac{b^2}{a+b} + 2bf(x_0) - bf(x_1) - \frac{3}{2}[f(x_1) - f(x_0)] \cdot \frac{a^2}{a+b} - 2af(x_0) + af(x_1)$$

$$= \left[\frac{3}{2} \cdot f(x_1) - \frac{3}{2} \cdot f(x_0) + 2f(x_0) - f(x_1)\right](b-a) = \left[\frac{1}{2}f(x_1) - \frac{1}{2}f(x_0)\right](b-a)$$

$$= \frac{1}{2}\left[f\left(\frac{2}{3}(a+b)\right) + f\left(\frac{1}{3}(a+b)\right)\right](b-a)$$

(C) $\int_0^1 e^x\,dx = e - 1 = 1.7183$

Trapezoid: $\approx \frac{1}{2}(e+1) = 1.8591$

Open: $\approx \frac{1}{2}(e^{\frac{2}{3}} + e^{\frac{1}{3}}) = 1.6717$

$\int_0^\pi \sin(x)\,dx = 1 + 1 = 2$

Trapezoid: $\approx \frac{\pi}{2} = 1.5708$

Open: $\approx \frac{\pi}{2}\left(\frac{\sqrt{3}}{2} + \frac{\sqrt{3}}{2}\right) = 2.7207$

③ Ⓐ $r_i = y^{(i)} - f(x_1^{(i)}, x_2^{(i)})$

$$= y^{(i)} - \begin{bmatrix} 1 & x_1^{(i)} & x_2^{(i)} & x_1^{(i)}x_2^{(i)} \end{bmatrix} \begin{bmatrix} \beta_0^* \\ \beta_1^* \\ \beta_2^* \\ \beta_3^* \end{bmatrix}$$

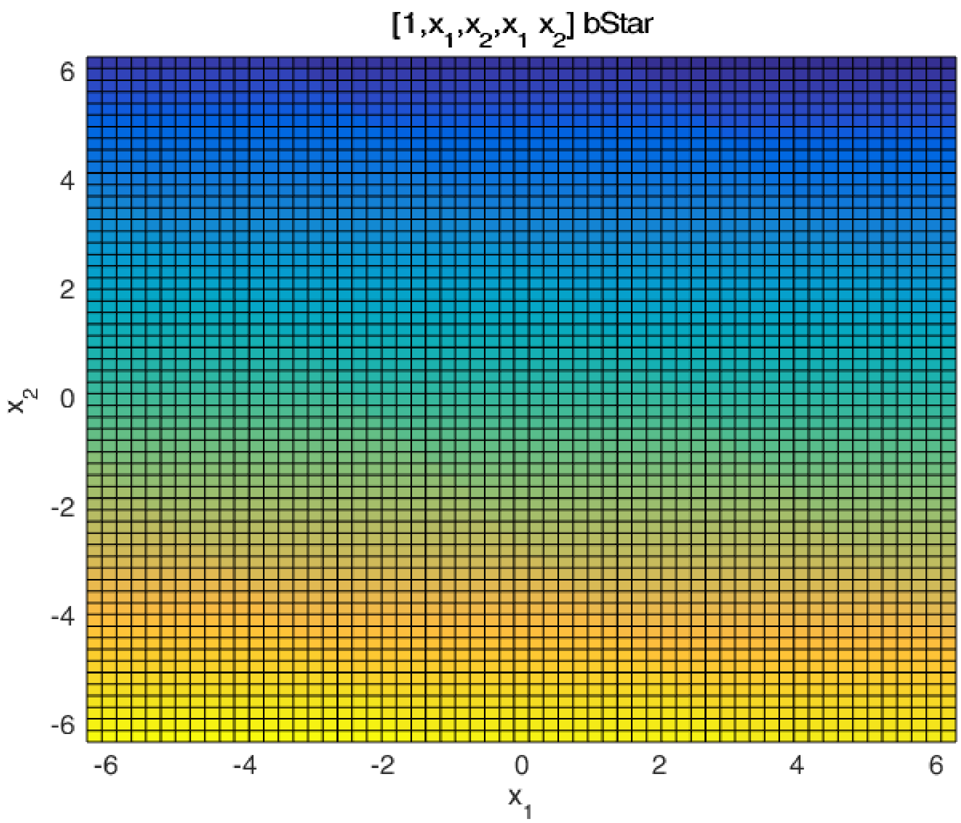
$$\Rightarrow \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix} = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} & x_1^{(1)}x_2^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & x_1^{(2)}x_2^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_1^{(m)} & x_2^{(m)} & x_1^{(m)}x_2^{(m)} \end{bmatrix} \begin{bmatrix} \beta_0^* \\ \beta_1^* \\ \beta_2^* \\ \beta_3^* \end{bmatrix}$$

$$r = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix} - \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} & x_1^{(1)}x_2^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & x_1^{(2)}x_2^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_1^{(m)} & x_2^{(m)} & x_1^{(m)}x_2^{(m)} \end{bmatrix} \begin{bmatrix} \beta_0^* \\ \beta_1^* \\ \beta_2^* \\ \beta_3^* \end{bmatrix}$$

Ⓑ Attached

```
%Problem 3
load carbig;
m = length(MPG);

% Get number of NaN data points
badCount = 0;
for i = 1:m
    if isnan(MPG(i)) || isnan(Weight(i)) || isnan(Horsepower(i))
        badCount = badCount + 1;
    end
end

n = m - badCount;
y = zeros(n, 1);
x1s = zeros(n, 1);
x2s = zeros(n, 1);
A = zeros(n, 4);

% Populate matrices with "good" data
i = 1;
for j = 1:m
    if isnan(MPG(j)) || isnan(Weight(j)) || isnan(Horsepower(j))
        continue;
    end
    y(i) = MPG(j);
    x1 = Weight(j);
    x1s(i) = x1;
    x2 = Horsepower(j);
    x2s(i) = x2;
    A(i, :) = [1, x1, x2, x1 * x2];
    i = i + 1;
end

% Part B
bStar = A\y;

% Part C
hold on;
scatter3(x1s, x2s, y);
title('Problem 3, part (c)');
fh = @(x1,x2) [1, x1, x2, x1 * x2] * bStar;
ezsurf(fh);
hold off;
```

Warning: Function failed to evaluate on array inputs; vectorizing the function
may speed up its evaluation and avoid the need to loop over array elements.
Warning: Function failed to evaluate on array inputs; vectorizing the function
may speed up its evaluation and avoid the need to loop over array elements.



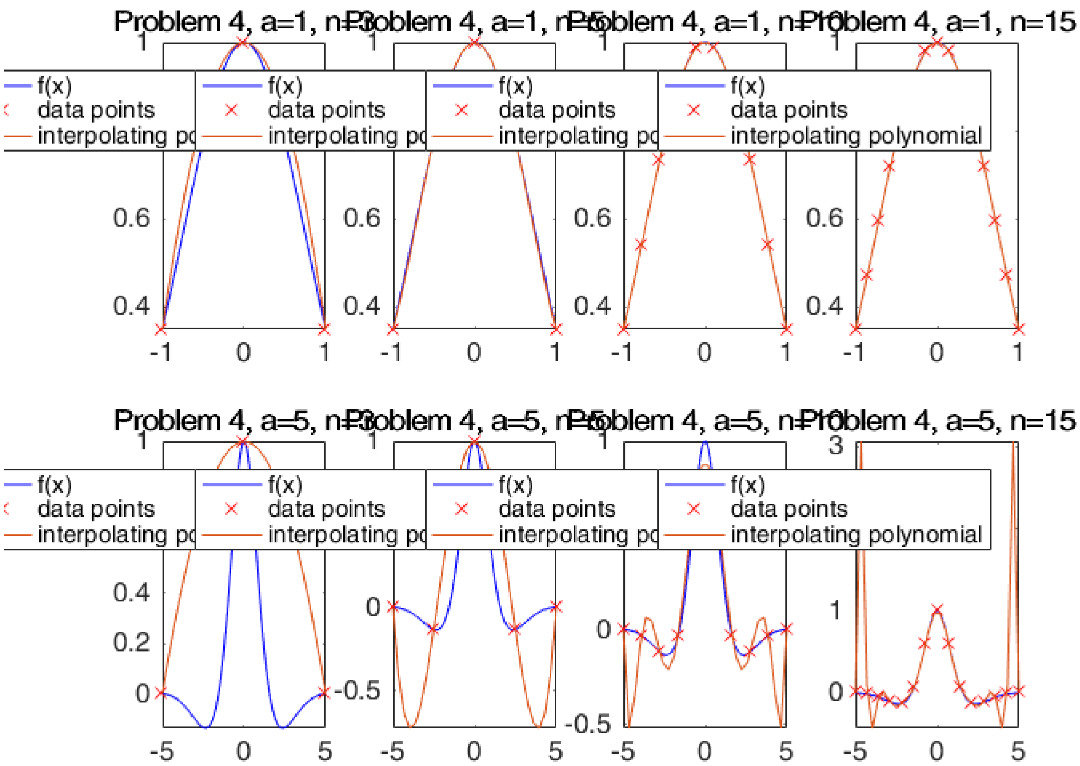
Published with MATLAB® R2016a

```
% Problem 4
f = @(x) cos(x)/cosh(x);

subplot_num = 1;
grid on;

a = 1;
n = 3;
[points, coefficients] = find_coefficients(a, n);
% note that find_coefficients is defined in a separate file
subplot(2, 4, subplot_num);
subplot_num = subplot_num + 1;
fplot(f, [-a a], '-b'); hold on; plot(points(:, 1), points(:,
 2), 'xr');
xs = linspace(-a, a, 30);
plot(xs, polyval(coefficients, xs)); hold off;
title(strcat(strcat('Problem 4, a=', int2str(a)), strcat(', n=',
 int2str(n))));
legend('f(x)', 'data points', 'interpolating polynomial');

a = 1;
n = 5;
[points, coefficients] = find_coefficients(a, n);
subplot(2, 4, subplot_num);
subplot_num = subplot_num + 1;
fplot(f, [-a a], '-b'); hold on; plot(points(:, 1), points(:,
 2), 'xr');
xs = linspace(-a, a, 30);
plot(xs, polyval(coefficients, xs)); hold off;
title(strcat(strcat('Problem 4, a=', int2str(a)), strcat(', n=',
 int2str(n))));
legend('f(x)', 'data points', 'interpolating polynomial');

a = 1;
n = 10;
[points, coefficients] = find_coefficients(a, n);
subplot(2, 4, subplot_num);
subplot_num = subplot_num + 1;
fplot(f, [-a a], '-b'); hold on; plot(points(:, 1), points(:,
 2), 'xr');
xs = linspace(-a, a, 30);
plot(xs, polyval(coefficients, xs)); hold off;
title(strcat(strcat('Problem 4, a=', int2str(a)), strcat(', n=',
 int2str(n))));
legend('f(x)', 'data points', 'interpolating polynomial');

a = 1;
n = 15;
[points, coefficients] = find_coefficients(a, n);
subplot(2, 4, subplot_num);
subplot_num = subplot_num + 1;
```

```
fplot(f, [-a a], '-b'); hold on; plot(points(:, 1), points(:,
 2), 'xr');
xs = linspace(-a, a, 30);
plot(xs, polyval(coefficients, xs)); hold off;
title(strcat(strcat('Problem 4, a=', int2str(a)), strcat(', n=',
 int2str(n))));
legend('f(x)', 'data points', 'interpolating polynomial');

a = 5;
n = 3;
[points, coefficients] = find_coefficients(a, n);
subplot(2, 4, subplot_num);
subplot_num = subplot_num + 1;
fplot(f, [-a a], '-b'); hold on; plot(points(:, 1), points(:,
 2), 'xr');
xs = linspace(-a, a, 30);
plot(xs, polyval(coefficients, xs)); hold off;
title(strcat(strcat('Problem 4, a=', int2str(a)), strcat(', n=',
 int2str(n))));
legend('f(x)', 'data points', 'interpolating polynomial');

a = 5;
n = 5;
[points, coefficients] = find_coefficients(a, n);
subplot(2, 4, subplot_num);
subplot_num = subplot_num + 1;
fplot(f, [-a a], '-b'); hold on; plot(points(:, 1), points(:,
 2), 'xr');
xs = linspace(-a, a, 30);
plot(xs, polyval(coefficients, xs)); hold off;
title(strcat(strcat('Problem 4, a=', int2str(a)), strcat(', n=',
 int2str(n))));
legend('f(x)', 'data points', 'interpolating polynomial');

a = 5;
n = 10;
[points, coefficients] = find_coefficients(a, n);
subplot(2, 4, subplot_num);
subplot_num = subplot_num + 1;
fplot(f, [-a a], '-b'); hold on; plot(points(:, 1), points(:,
 2), 'xr');
xs = linspace(-a, a, 30);
plot(xs, polyval(coefficients, xs)); hold off;
title(strcat(strcat('Problem 4, a=', int2str(a)), strcat(', n=',
 int2str(n))));
legend('f(x)', 'data points', 'interpolating polynomial');

a = 5;
n = 15;
[points, coefficients] = find_coefficients(a, n);
subplot(2, 4, subplot_num);
subplot_num = subplot_num + 1;
fplot(f, [-a a], '-b'); hold on; plot(points(:, 1), points(:,
 2), 'xr');
```

```
xs = linspace(-a, a, 30);
plot(xs, polyval(coefficients, xs)); hold off;
title(strcat(strcat('Problem 4, a=', int2str(a)), strcat(', n=',
 int2str(n))));
legend('f(x)', 'data points', 'interpolating polynomial');
```

Warning: Function fails on array inputs. Use element-wise operators to increase speed.
Warning: Function fails on array inputs. Use element-wise operators to increase speed.
Warning: Function fails on array inputs. Use element-wise operators to increase speed.
Warning: Function fails on array inputs. Use element-wise operators to increase speed.
Warning: Function fails on array inputs. Use element-wise operators to increase speed.
Warning: Function fails on array inputs. Use element-wise operators to increase speed.
Warning: Function fails on array inputs. Use element-wise operators to increase speed.
Warning: Polynomial is badly conditioned. Add points with distinct X values, reduce the degree of the polynomial, or try centering and scaling as described in HELP POLYFIT.
Warning: Function fails on array inputs. Use element-wise operators to increase speed.



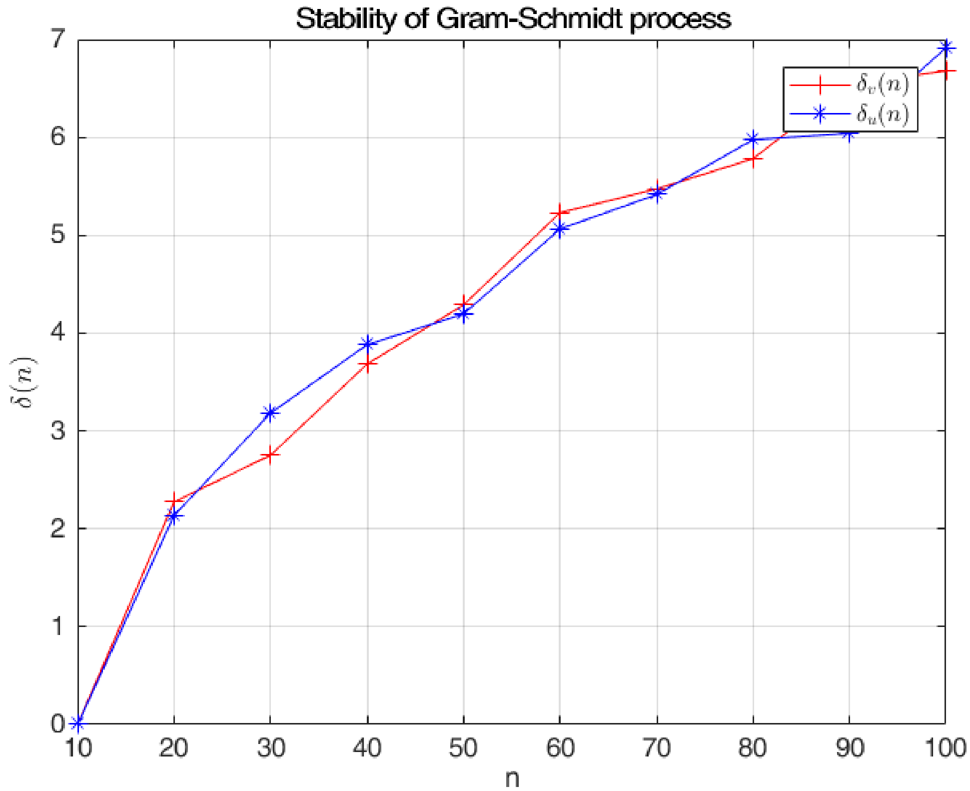
Published with MATLAB® R2016a

```
% Problem 4 helper

function [points, coefficients] = find_coefficients(a, n)
    f = @(x) cos(x)/cosh(x);
    points = zeros(n, 2);
    points(:, 1) = linspace(-a, a, n);
    for i = 1:n
        points(i, 2) = f(points(i, 1));
    end
    coefficients = polyfit(points(:, 1), points(:, 2), n - 1);
end
```

```
Not enough input arguments.

Error in find_coefficients (line 5)
    points = zeros(n, 2);
```

Published with MATLAB® R2016a

```matlab
% Problem 5

clear deltav;
clear deltau;
ns = 10:10:100;
deltav = zeros(10, 1);
deltau = zeros(10, 1);
for n = ns
    % gram_schmidt is defined in a separate file
    V = gram_schmidt(hilb(n));
    A1 = eye(n) - V' * V;
    deltav(n/10) = norm(A1, Inf);
    % gram_schmidt_stable is defined in a separate file
    U = gram_schmidt_stable(hilb(n));
    A2 = eye(n) - U' * U;
    deltau(n/10) = norm(A2, Inf);
end

plot(ns, deltav, '-+r');
title('Stability of Gram-Schmidt process');
grid on;
hold on;
plot(ns, deltau, '-*b');
legend({'$\delta_v(n)$', '$\delta_u(n)$'},'Interpreter','latex');
ylabel('$\delta(n)$','Interpreter','latex');
xlabel('n');
hold off;
```



Published with MATLAB® R2016a

```
% Problem 5 helper 1

function V = gram_schmidt(H)
    n = size(H);
    V = zeros(n);
    % Gram-Schmidt process
    for i = 1:n
        v = H(:, i);
        for j = 1:i-1
            vj = V(:, j);
            v = v - (dot(v, vj)/power(norm(vj), 2)) * vj;
        end
        V(:, i) = v;
    end

    for i = 1:n
        u = V(:, i);
        V(:, i) = u/norm(u);
    end
end
```

```
Not enough input arguments.

Error in gram_schmidt (line 4)
    n = size(H);
```

Published with MATLAB® R2016a

```
%Problem 5 helper 2
function U = gram_schmidt_stable(H)
    n = size(H);
    U = zeros(n);

    % Gram-Schmidt process
    for i = 1:n
        u = H(:, i) / norm(H(:, i));
        U(:, i) = u;
        for j = i + 1:n
            H(:, j) = H(:, j) - dot(H(:, j), u) * u;
        end
    end
end
```

```
Not enough input arguments.

Error in gram_schmidt_stable (line 3)
    n = size(H);
```

Published with MATLAB® R2016a